Apprentissage de dictionnaire non paramétrique pour les problèmes inverses en traitement d'image

Hong-Phuong Dang[1] & Pierre Chainais[2]

[1] *ENSAI 35172 Bruz cedex, UMR 9194 - CREST, France - hong-phuong.dang@ensai.fr*
[2] *Uni. Lille, Centrale Lille, UMR 9189 - CRIStAL, France - pierre.chainais@ec-lille.fr*

Résumé. L'apprentissage de dictionnaire pour la représentation parcimonieuse est bien connu pour résoudre des problèmes inverses en traitement d'image. Généralement, le nombre d'atomes du dictionnaire est fixé à l'avance. En utilisant un modèle Bayésien non paramétrique de type Buffet Indien, nous proposons une méthode qui apprend automatiquement un dictionnaire de taille adaptée. Le niveau de bruit est aussi estimé avec précision, de sorte que presqu'aucun réglage des paramètres n'est nécessaire. Les résultats comparatifs de débruitage, inpainting et acquisition compressée illustrent la pertinence de la méthode proposée.

Mots-clés. représentation parcimonieuse, apprentissage de dictionnaire, problème inverse, bayésien non paramétrique, processus du Buffet Indien, Monte-Carlo par chaînes de Markov, ...

Abstract. Dictionary learning for sparse representation is well known in solving inverse problems in image processing. In general, the number of dictionary atoms is fixed in advance. By using a Bayesian non parametric approach : the Indian Buffet Process prior, we propose a dictionary learning approach that automatically learns a dictionary of adapted size. The noise level is also accurately estimated so that nearly no parameter tuning is needed. The denoising, inpainting and compressed sensing comparative results show the relevance of the proposed method.

Keywords. sparse representation, dictionary learning, inverse problem, Bayesian nonparametric, Indian Buffet Process, Markov chain Monte Carlo, ...

1 Introduction

L'*apprentissage de dictionnaire* (AD) est une méthode éprouvée pour la résolution *problèmes inverses* (débruitage, inpainting, acquisition compressée, ...). Ce sont souvent des *problèmes mal posés* que l'on régularise avec des contraintes de parcimonie. Beaucoup d'attention a été portée aux méthodes d'optimisation (par ex. Aharon *et al* (2006), Mairal *et al* (2010), Rao *et al* (2006)) où le dictionnaire optimal est obtenu à partir d'un critère de parcimonie donné pour une image. Dans ces méthodes d'optimisation, la parcimonie est typiquement favorisée par une pénalité de type ℓ_0 ou ℓ_1 sur l'ensemble des cœfficients

du code et la taille du dictionnaire est fixée à l'avance. À l'inverse, l'AD par des approches bayésiennes a été étudié dans une moindre mesure. Zhou *et al* (2012) présente une approche de cette famille en introduisant une loi *a priori* Bêta-Bernoulli sur le support des représentations pour favoriser la parcimonie. Malgré des connexions avec les approches bayésiennes non paramétriques, elle fonctionne encore avec un dictionnaire de taille fixe.

Le fait de fixer la dimension des paramètres qui peut être difficile à spécifier constitue une limite des modèles paramétriques. Les modèles non paramétriques considèrent que le dictionnaire est de taille potentiellement infinie mais un a priori favorisant la parcimonie de la représentation est introduit. Cela permet de développer des méthodes d'AD sans fixer à l'avance la taille du dictionnaire. Ce papier présente un modèle *bayésien non paramétrique* (BNP), sans fixer à l'avance la taille du dictionnaire grâce à l'utilisation d'une loi *a priori* nommée *Processus du Buffet Indien* (IBP).

La partie 2 rappelle le principe de l'apprentissage de dictionnaire. La partie 3 présente le processus Buffet Indien et le modèle proposé avant de décrire l'échantillonnage de Gibbs pour l'inférence. La partie 4 est consacrée aux résultats expérimentaux et à la discussion.

2 Apprentissage de dictionnaire (AD)

En traitement d'image, chaque observation est en général une imagette (*patch cf.* Tosic et Frossard (2011)) carrée (par ex. 8×8) rangée dans un vecteur de dimension L (par ex. 64) par ordre lexicographique. L'observation est le résultat d'une fonction éventuellement non linéaire. En particulier, dans le cas linéaire on a :

$$\mathbf{Y} = \mathbf{H}(\mathbf{X} + \boldsymbol{\varepsilon}) \quad \text{où} \quad \mathbf{X} = \mathbf{D}\mathbf{W}. \tag{1}$$

$\mathbf{X}$ contient les patches de l'image initiale qui est perturbée par un opérateur linéaire $\mathbf{H}$ connu et un bruit $\boldsymbol{\varepsilon}$. Cela donne $\mathbf{Y} = [\mathbf{y}_1, ..., \mathbf{y}_N] \in \mathbb{R}^{L\times N}$ qui est l'ensemble des N patches extraits d'une image observable. On suppose dans la suite que le bruit $\mathbf{H}\boldsymbol{\varepsilon}$ est gaussien i.i.d.. On cherche à retrouver $\mathbf{X}$ à partir de $\mathbf{Y}$. Pour cela, les patches sont représentés par les coefficients de codage $\mathbf{W}=[\mathbf{w}_1, ..., \mathbf{w}_N]\in\mathbb{R}^{K\times N}$ de leur représentation dans un dictionnaire $\mathbf{D}=[\mathbf{d}_1, ..., \mathbf{d}_K]\in\mathbb{R}^{L\times K}$ avec K atomes. Chaque $\mathbf{x}_i$ est décrit par $\mathbf{x}_i = \mathbf{D}\mathbf{w}_i$ où $\mathbf{w}_i$ est épars. La récupération de $\mathbf{X}$ est équivalente dans un certain sens à la recherche d'un couple optimal $(\mathbf{D}, \mathbf{W})$.

On peut aborder la question comme un problème d'optimisation jointe où la parcimonie est typiquement imposée par une pénalité ℓ_0 ou ℓ_1 :

$$(\mathbf{D}, \mathbf{W}) = \underset{(\mathbf{D},\mathbf{W})}{\operatorname{argmin}} \quad \frac{1}{2}\|\mathbf{H}(\mathbf{Y} - \mathbf{D}\mathbf{W})\|_2^2 + \lambda\|\mathbf{W}\|_p, \quad p \in \{0, 1\}. \tag{2}$$

La taille de dictionnaire est le plus souvent fixée à l'avance à K=256 ou 512 atomes.

Dans le cadre bayésien, le problème est traduit dans une vraisemblance gaussienne selon le modèle (1). La loi *a priori* $\mathrm{p}(\mathbf{D}, \mathbf{W}, \sigma_{\boldsymbol{\varepsilon}})$ agit comme une régularisation et la distribution *a posteriori* jointe s'écrit :

$$p(\mathbf{D}, \mathbf{W}, \sigma_{\boldsymbol{\varepsilon}} \mid \mathbf{Y}, \mathbf{H}) \propto p(\mathbf{Y} \mid \mathbf{H}, \mathbf{D}, \mathbf{W}, \sigma_{\boldsymbol{\varepsilon}})p(\mathbf{D}, \mathbf{W}, \sigma_{\boldsymbol{\varepsilon}}). \tag{3}$$

En utilisant par exemple l'échantillonnage de Gibbs pour l'inférence, le problème peut être résolu en échantillonnant alternativement $\mathbf{D}$, $\mathbf{W}$ et $\sigma_{\boldsymbol{\varepsilon}}$. Dans un cadre BNP, le dictionnaire est appris sans réglage préalable de la taille et aucun réglage de paramètre n'est nécessaire. Le modèle IBP-DL présenté dans 3.2 a utilisé le processus du buffet indien pour traiter à la fois la contrainte de parcimonie et le nombre adaptatif souhaitable d'atomes.

3 Apprentissage de dictionnaire non paramétrique

3.1 Processus du Buffet Indien (IBP)

Conformément à l'équation (1), le nombre d'atomes est non seulement défini par le nombre de colonnes de $\mathbf{D}$ mais aussi par le nombre de lignes de $\mathbf{W}$. Lorsqu'une ligne de $\mathbf{W}$ ne contient que des zéros, cela revient à supprimer de $\mathbf{D}$ la colonne de l'atome correspondant à cette ligne de $\mathbf{W}$. La parcimonie de $\mathbf{W}$ est contrôlée par un support $\mathbf{Z}$. Cette matrice binaire $\mathbf{Z}$ modélise l'utilisation des éléments d'un dictionnaire par des données dans un modèle à caractéristiques latentes (*Latent Features*). Si la donnée i est associée à la caractéristique k alors $\mathbf{Z}(k,i)$=1, (0 si non). Chaque donnée peut posséder un ensemble de caractéristiques. Dans le cas où le nombre de caractéristiques utilisées est inconnu, l'IBP introduit par Griffiths et Ghahramani (2005) permet de définir une loi *a priori* sur $\mathbf{Z}$ de taille (nombre de lignes) potentiellement infinie. Il existe plusieurs façon pour présenter l'IBP. Thibaux et Jordan (2007) ont montré que l'IBP est obtenu par intégration d'un processus Bêta-Bernoulli vis-à-vis du processus Bêta, faisant de ce processus la mesure de Finetti sous-jacente à l'IBP. Historiquement, L'IBP est plutôt présenté à travers la métaphore du Buffet Indien. Chaque client (donnée) i choisit d'abord le plat (caractéristique) k avec probabilité m_k/i où m_k est le nombre de fois où le plat k a été choisi par les clients précédents. Puis ce client i choisit encore $k_{new} \sim \text{Poisson}(\alpha/i)$ nouveaux plats. Cette étape permet d'enrichir progressivement le dictionnaire. L'IBP est caractérisé par une distribution sur les classes d'équivalence de matrices binaires :

$$P([\mathbf{Z}]) = \frac{1}{\prod_{h=1}^{2^N-1} K_h!} \exp(-\alpha H_N) \prod_{k=1}^{\mathrm{K}} \frac{(N-m_k)!(m_k-1)!}{N!} \quad \text{où} \quad H_N = \sum_{i=1}^{N} \frac{1}{i} \tag{4}$$

où N le nombre de données ou encore le nombre de colonnes de $\mathbf{Z}$, $m_k = \sum_{i=1}^{N} \mathbf{Z}(i,k)$ le nombre de données utilisant l'atome k, K le nombre d'atomes "actifs" tels que m_k>0, K_h est le nombre d'atomes avec le même *histoire* $\mathbf{Z}(:,k)$=h. Autrement dit, les plats ont été choisis par la même ensemble de client. Le paramètre α>0 de l'IBP quantifie le niveau de régularisation puisque $\mathbb{E}[K_+] = \alpha H_N \asymp \alpha \log N$. Plus α est petit, plus la régularisation est forte. En bref, l'IBP génère des matrices binaires *creuses* et *potentiellement infinies.*

3.2 Modèle IBP-DL

L'acronyme IBP-DL signifie Indian Buffet Process for Dictionary Learning. La figure 1 montre le modèle graphique de l'IBP-DL. $\odot$ est le produit Hadamard (terme à terme).

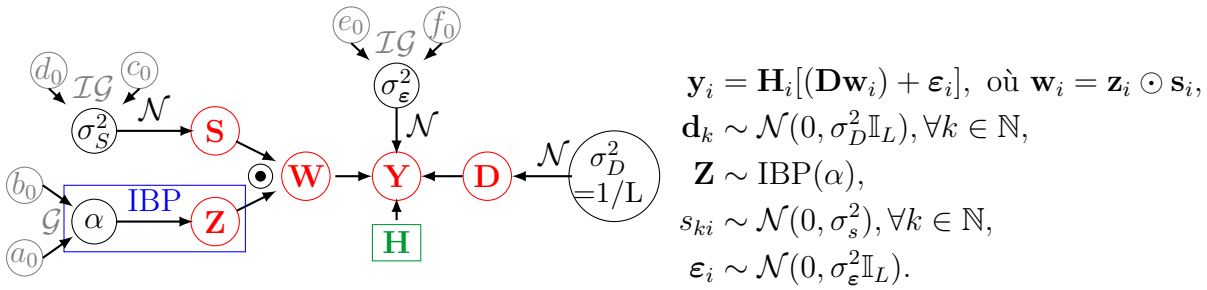

FIGURE 1 – IBP-DL pour les problèmes inverses linéaires avec un bruit gaussien.

Dans l'esprit d'un modèle Bernoulli-gaussien paramétrique, la parcimonie de $\mathbf{W}$ est induite par celle de la matrice binaire $\mathbf{Z}$ grâce à l'*a priori* non paramétriques IBP. Si $\mathbf{Z}(k,i) = 1$, la donnée i utilise l'atome k alors $\mathbf{W}(k,i) = \mathbf{S}(k,i)$, (0 si non). De plus, cette loi contrôle le nombre de lignes de $\mathbf{Z}$ (ou encore $\mathbf{W}$) qui représente aussi le nombre d'atomes de $\mathbf{D}$. Enfin, l'amplitude du code $\mathbf{W}$ est donnée par $\mathbf{S}$. Des lois *a priori* conjuguées inverse Gamma et Gamma sont utilisées pour les paramètres $\sigma_S^2, \sigma_\varepsilon^2$, les variances des lois Normales et α, paramètre associé à une loi de Poisson dans l'IBP. Seule la variance σ_D^2 est fixée à $1/L$ pour régler le problème d'indétermination liée à la norme du couple ($\mathbf{D}$, $\mathbf{W}$). Si $\mathbf{H}_i = \mathbb{I}_L$, il s'agit d'un problème de débruitage. Si $\mathbf{H}_i$ est une matrice binaire diagonale, on est dans le cas de l'inpainting. Dans le cas de l'acquisition compressée, $\mathbf{H}_i \in \mathbb{R}^{Q \times L}$ où $Q < L$.

3.3 Algorithme pour inférer IBP-DL

Init. : K=0, $\mathbf{Z} = \mathbf{S} = \mathbf{W} = \emptyset$, $\mathbf{D} = \emptyset$, α=1, $\sigma_D^2 = L^{-1}$, σ_S^2=1, σ_ε
Résultats : $\mathbf{D} \in \mathbb{R}^{P \times K}, \mathbf{Z} \in \{0;1\}^{K \times P}, \mathbf{S} \in \mathbb{R}^{K \times P}$, σ_ε
Pour *chaque itération t*
- Utiliser les statistiques suffisantes selon éq.(6)
- **Pour** *chaque donnée i=1 :N*
  - Enlever l'influence de donnée i via éq.(7)
  - $m_{-i} \in \mathbb{N}^{K \times 1} \leftarrow \sum(\mathbf{Z}(:, -i))$
  - **Pour** $k \in k_{used} \leftarrow find(m_{-i} \neq 0)$
    - Inférer $\mathbf{Z}(k, , i)$ selon eq.(5) et (8)
  - Inférer nouveaux atomes par Metropolis-Hastings (c.f Dang et Chainais (2016, 2017))
  - Remettre l'influence de donnée i via éq.(7)
  - **Pour** *chaque atome k=1 :K*
    - Échantillonner $\mathbf{d}_k$ éq. (9)
    - Échantillonner $\mathbf{s}_k(\mathbf{z}_k \neq 0)$ éq. (10)
- Échantillonner σ_S, σ_ε,α

Algo. 1: L'échantillonneur de Gibbs marginalisé accéléré pour IBP-DL.

Différentes méthodes d'échantillonnage sont étudiées pour IBP-DL. Cette section décrit brièvement la stratégie pour échantillonner la distribution *a posteriori* $\text{p}(\mathbf{D}, \mathbf{S}, \mathbf{Z}, ...|\mathbf{Y}, \mathbf{H})$

(voir Algo. 1). On présente ici une version de l'échantillonneur de Gibbs marginalisé accéléré pour $\mathbf{Z}$ pour l'inpainting, à base de statistiques suffisantes. Les nouveaux atomes sont inférés grâce à une étape Metropolis-Hastings (MH).

La mise à jour $z_{ki} = \mathbf{Z}(k,i)$ pour les atomes k existant, dits "actifs", est réalisée via :

$$\begin{aligned}\mathrm{P}(z_{ki}=1 \mid \mathbf{Y},\mathbf{H},\mathbf{W},\sigma_D,\sigma_{\boldsymbol{\varepsilon}},\alpha) &\propto \frac{m_{k,-i}}{N} \times \int \mathrm{p}(\mathbf{y}_i \mid \mathbf{H}_i,\mathbf{w}_i,\mathbf{D}) p(\mathbf{Y}_{-i} \mid \mathbf{H}_{\neq i},\mathbf{W}_{-i},\mathbf{D}) \mathrm{p}(\mathbf{D} \mid \sigma_D) d\mathbf{D} \\ &\propto \frac{m_{k,-i}}{N} \int p(\mathbf{y}_i \mid \mathbf{H}_i,\mathbf{D},\mathbf{w}_i) \prod_{\ell=1}^{L} \mathrm{p}(\mathbf{D}(\ell,:) \mid \mathbf{F}_\ell,\mathbf{W}_{-i},\sigma_D) d\mathbf{D} \quad (5)\end{aligned}$$

Les données $\mathbf{Y}$ et la matrice des coefficients $\mathbf{W}$ peuvent se répartir en 2 parties, une partie pour la donnée i, l'autre pour le reste. $\{\mathbf{F}_\ell\}_{\ell=1,\dots,L}$ est l'ensemble des matrices binaires diagonales de taille N. $\mathbf{F}_\ell(i,i)$=$\mathbf{H}_i(\ell,\ell)$ indique si le pixel à l'emplacement ℓ du patch i est observé ou non. On peut montrer que la loi *a posteriori* de $\mathbf{D}$ est normalement distribuée selon une espérance $\boldsymbol{\mu}_{\mathbf{D}\ell}$ et une covariance $\boldsymbol{\Sigma}_{\mathbf{D}\ell}$. L'idée est de travailler sur $\mathbf{D}$ ligne par ligne (dimension) plutôt que par colonnes (atomes). On utilise la notion de *statistiques suffisantes (information form)* pour modéliser l'influence d'une observation i à part dans la loi *a posteriori* de $\mathbf{D}$:

$$g_{\mathbf{D}\ell} = \boldsymbol{\Sigma}_{\mathbf{D}\ell}^{-1} = (1/\sigma_{\boldsymbol{\varepsilon}}^2)\mathbf{M}_\ell^{-1} \quad \text{et} \quad h_{\mathbf{D}\ell} = \boldsymbol{\mu}_{\mathbf{D}\ell} g_{\mathbf{D}\ell} = (1/\sigma_{\boldsymbol{\varepsilon}}^2)\mathbf{Y}(\ell,:)\mathbf{F}_\ell^T\mathbf{W}^T \quad (6)$$

avec $\mathbf{M}_\ell = (\mathbf{W}\mathbf{F}_\ell\mathbf{F}_\ell^T\mathbf{W}^T + \frac{\sigma_{\boldsymbol{\varepsilon}}^2}{\sigma_D^2}\mathbb{I}_K)^{-1}$. En effet, on peut définir :

$$g_{\mathbf{D}\ell,\pm i} = g_{\mathbf{D}\ell} \pm \sigma_{\boldsymbol{\varepsilon}}^{-2} H_{i,\ell}\ \mathbf{w}_i\mathbf{w}_i^{\mathrm{T}} \quad \text{et} \quad h_{\mathbf{D}\ell,\pm i} = h_{\mathbf{D}\ell} \pm \sigma_{\boldsymbol{\varepsilon}}^{-2} H_{i,\ell}\ y_i(\ell)\ \mathbf{w}_i^{\mathrm{T}} \quad (7)$$

Comme la vraisemblance est aussi gaussienne, l'intégrale (5) est proportionnelle à :

$$\prod_{\ell=1}^{L} \mathcal{N}(y_i(\ell); H_{i,\ell}\boldsymbol{\mu}_{\mathbf{D}\ell,-i}\mathbf{w}_i, H_{i,\ell}\mathbf{w}_i^{\mathrm{T}}\boldsymbol{\Sigma}_{\mathbf{D}\ell,-i}\mathbf{w}_i + \sigma_{\boldsymbol{\varepsilon}}^2) \quad (8)$$

L'algorithme Métropolis-Hastings prenant en compte des cas *singletons* est utilisé pour activer les nouveaux atomes afin de respecter les hypothèses d'un échantillonnage de Gibbs correct. Les "singletons" sont des cas où un plat est utilisé par un seul client.

$\mathbf{D}$ et $\mathbf{S}$ sont échantillonnés selon les distributions gaussiennes :

$$\mathbf{d}_k|\mathbf{Y},\mathbf{H},\mathbf{W},\mathbf{D}_{-k},\dots \sim \mathcal{N}(\boldsymbol{\mu}_{\mathbf{d}_k},\boldsymbol{\Sigma}_{\mathbf{d}_k}) \quad (9)$$

où $\boldsymbol{\Sigma}_{\mathbf{d}_k} = (\sigma_D^{-2}\mathbb{I}_L + \sigma_{\boldsymbol{\varepsilon}}^{-2}\sum_{i=1}^{N} w_{ki}^2\mathbf{H}_i^T\mathbf{H}_i)^{-1}$ et $\boldsymbol{\mu}_{\mathbf{d}_k} = \sigma_{\boldsymbol{\varepsilon}}^{-2}\boldsymbol{\Sigma}_{\mathbf{d}_k}\sum_{i=1}^{N} w_{ki}(\mathbf{H}_i^T\mathbf{y}_i - \mathbf{H}_i^T\mathbf{H}_i\sum_{j\neq k}^{K}\mathbf{d}_j w_{ji})$

$$s_{ki}|\mathbf{Y},\mathbf{H}_i,\mathbf{D},\mathbf{Z},\mathbf{S}_{k,-i},\dots \sim \mathcal{N}(\mu_{s_{ki}},\Sigma_{s_{ki}}) \quad (10)$$

$$z_{ki}=1 \Rightarrow \begin{cases} \Sigma_{s_{ki}} = (\sigma_{\boldsymbol{\varepsilon}}^{-2}\mathbf{d}_k^T\mathbf{H}_i^T\mathbf{H}_i\mathbf{d}_k + \sigma_S^{-2})^{-1} \\ \mu_{s_{ki}} = \sigma_{\boldsymbol{\varepsilon}}^{-2}\Sigma_{s_{ki}}\mathbf{d}_k^T(\mathbf{H}_i^T\mathbf{y}_i - \mathbf{H}_i^T\mathbf{H}_i\sum_{j\neq k}^{K}\mathbf{d}_j w_{ji}) \end{cases} ; \quad z_{ki}=0 \Rightarrow \begin{cases} \Sigma_{s_{ki}} = \sigma_S^2 \\ \mu_{s_{ki}} = 0 \end{cases}$$

Les paramètres $\boldsymbol{\theta} = \sigma_{\boldsymbol{\varepsilon}}$, σ_S et α sont aussi échantillonnés (c.f Dang et Chainais (2017)).

4 Résultats et discussions

Dang et Chainais (2016) ont présenté des résultats de débruitage de IBP-DL. Dang et Chainais (2017) ont aussi montré la pertinence de l'IBP-DL pour l'inpainting ainsi que l'acquisition compressée. Les performances numériques sont comparables aux autres approches AD tout en étant non paramétriques. Notez que l'erreur d'estimation sur le niveau de bruit est au maximum de 10% dans ce cas. On retient aussi que le nombre d'atomes K appris est expérimentalement souvent plus petit ou légèrement supérieur à la dimension des données (64 ici). Cette observation est surprenante car à contre courant des méthodes paramétriques qui fixent la taille du dictionnaire à un nombre grand devant la dimension (typiquement 256). Toutefois, les résultats numériques laissent présager que la taille des dictionnaires n'est pas critique vis-à-vis des performances de restauration.

L'intérêt des méthodes BNP est de plus en plus reconnu pour les méthodes d'apprentissage statistiques (machine learning). La méthode IBP-DL proposée permettant d'apprendre un dictionnaire de taille adaptative. Cette méthode présente de plus l'avantage de ne nécessiter pratiquement aucun ajustement de paramètre. Une des limites de l'algorithme est son coût de calcul dû à l'échantillonneur de Gibbs. D'autre type d'inférence sont à l'étude pour réduire le temps de calcul.

Le lecteur intéressé est référé à Dang et Chainais (2017) pour plus de détails sur IBP-DL et les différents résultats. Dans une démarche de recherche reproductible, les codes Matlab et C sont mis à disposition sur `http://www.hongphuong-dang.com`

Bibliographie

[1] Aharon, M., Elad, M., et Bruckstein, A. (2006), K-SVD : An algorithm for designing overcomplete dictionaries for sparse representation, *IEEE Trans. Signal Process.*, 54 :4311– 4322.

[2] Dang, H.-P. et Chainais, P. (2016), Towards Dictionaries of Optimal Size : A Bayesian Non Parametric Approach, *Jour. of Signal Process. Systems*, 1–6.

[3] Dang, H.-P. et Chainais, P. (2017), Indian buffet process dictionary learning : algorithms and applications to image processing, *International Journal of Approximate Reasoning*.

[4] Griffiths, T. et Ghahramani, Z. (2005), Infinite latent feature models and the indian buffet process, *Advances in NIPS*.

[5] Mairal,J., Bach,F., Ponce,J., et Sapiro,G. (2010), Onlinel earning for matrix factorization and sparse coding, *J. Mach. Learn. Res.*, 11 :19–60.

[6] Rao, N. et Porikli, F. (2012), A clustering approach to optimize online dictionary learning, *IEEE ICASSP*, Kyoto.

[7] Thibaux, R. et Jordan, M. I. (2007), Hierarchical beta processes and the indian buffet process, *Int. Workshop on AISTATS* volume 11, pages 564–571.

[8] Tosic, I. et Frossard, P. (2011), Dictionary learning : What is the right representation for my signal, *IEEE Signal Process. Mag.*, 28 :27–38.

[9] Zhou, et al. (2012), Nonparametric bayesian dictionary learning for analysis of noisy and incomplete images, *IEEE Trans. Image Process.*, 21 :130–144.